2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)
August 23-27, 2021. Lyon, France

A Hybrid Filtered Basis Functions Approach for Tracking Control of Linear Systems with Unmodeled Nonlinear Dynamics

Cheng-Hao Chou, Molong Duan, and Chinedum E. Okwudire, *Member, IEEE*

2021 IEEE 17th International Conference on Automation Science and Engineering (CASE) | 978-1-6654-1873-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/CASE49439.2021.9551476

Abstract— **A hybrid filtered basis function (FBF) approach is proposed in this paper for feedforward tracking control of linear systems with unmodeled nonlinear dynamics. Unlike most available tracking control techniques, the FBF approach is very versatile; it is applicable to any type of linear system, regardless of its underlying dynamics. The FBF approach expresses the control input to a system as a linear combination of basis functions with unknown coefficients. The basis functions are forward filtered through a linear model of the system's dynamics and the unknown coefficients are selected such that tracking error is minimized. The linear models used in existing implementations of the FBF approach are typically physics-based representations of the linear dynamics of a system. The proposed hybrid FBF approach expands the application of the FBF approach to systems with unmodeled nonlinearities by learning from data. A hybrid model is formulated by combining a physics-based model of the system's linear dynamics with a data-driven linear model that approximates the unmodeled nonlinear dynamics. The hybrid model is used online in receding horizon to compute optimal control commands that minimize tracking errors. The proposed hybrid FBF approach is shown in simulations on a model of a vibration-prone 3D printer to improve tracking accuracy by up to 65.4%, compared to an existing FBF approach that does not incorporate data.**

I. Introduction

Tracking control is important in a wide range of automated systems. It aims at forcing a system's output to follow a defined reference by minimizing the tracking error, i.e., the error between the output and the reference. Tracking control can be performed using either feedforward (FF) or feedback (FB) approaches. Compared to FB approaches, which perform compensation after measuring errors, FF methods have the advantage of being able to pre-emptively cancel out the tracking errors using the system's model, and thus yield theoretically zero tracking error.

Perfect FF tracking control can be achieved by designing the controller to be the exact inverse of the system dynamics. However, exact inversion of system dynamics is usually problematic due to the uncancellable zeroes in the system dynamics [1]–[3]. Therefore, various FF controllers have been proposed in the literature to realize approximate model inversion. They include the zero magnitude error tracking controller (ZMETC), zero phase error tracking controller (ZPETC) [4], [5], extended bandwidth ZPETC [6], model matching [7], and the filtered basis functions (FBF) approach [8], [9], etc. Of the existing methods, the FBF approach stands out because of its versatility. It can handle any linear dynamics with excellent tracking performance [8]; it is also capable of minimizing control effort and enhancing robustness by choosing or designing different basis functions [10]–[12]. The FBF approach expresses the control input as a linear combination of basis functions with unknown coefficients. The basis functions are forward filtered through a linear model of the system's dynamics, and the unknown coefficients are selected such that tracking error is minimized [8]. A version of FBF, named filtered B-spline (FBS), uses B-splines as the basis to enable the implementation of the FBF approach in small batches (via receding horizon) [9]. This reduces the computational burden of the FBF approach and allows its online implementation for tracking lengthy reference signals. The FBS approach was implemented on a 3D printer in [9], leading to significant reductions in vibration-induced tracking errors. However, the existing FBF approaches, like other linear FF controllers, cannot handle nonlinearity or other unmodeled dynamics. The nonlinearity and unmodeled dynamics are prevalent in practical applications, such as friction in 3D printers. Hence, when applied to systems with significant unmodeled nonlinearity, the performance of the FBF controller degrades significantly.

With the growing availability of data in automated systems, methods that combine physics-based and data-driven (e.g., machine learning) approaches are gaining attention [13], [14]. For example, low-cost accelerometers could be added to the 3D printer studied in [9] to gather data online about its vibration. When data is available, a common approach used in tracking control is to tune or adapt the parameters of a linear physics-based model online using the data [15]–[18]. However, these approaches are unable to incorporate unmodeled dynamics that are non-parametric. Besides, several of these methods are based on iterative learning [16], [17], and hence, are not applicable to non-repeating trajectories.

To address these weaknesses of existing FF techniques, the authors have recently proposed a linear hybrid model, comprising a physics-based and data-driven model, for linear systems with unmodeled nonlinear dynamics [19]. The linear portion of the system's dynamics was modeled using the physics-based component of the hybrid model, while the unmodeled nonlinear portion was derived from its data-driven component. The hybrid model was shown in simulations and experiments to provide significantly improved predictions of servo errors in motion systems with unmodeled nonlinearity. However, the hybrid model was only used for error prediction but not for control. Therefore, as its primary contributions, this paper:

*Research supported by Grant # 1931950 from the National Science Foundation

C.-H. Chou (e-mail: cchengha@umich.edu) and C. E. Okwudire (e-mail: okwudire@umich.edu) are with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109 USA.

M. Duan (e-mail: duan@ust.hk) was with the Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109 USA. He is now with the Department of Mechanical and Aerospace Engineering, Hong Kong University of Science and Technology, Hong Kong, China.

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on December 29,2021 at 16:07:02 UTC from IEEE Xplore. Restrictions apply.

1. Proposes a hybrid FBF controller that uses a combination of physics-based and data-driven linear models (i.e., a linear hybrid model) to minimize tracking errors via FF control performed online in receding horizons.
2. Demonstrates in simulations (using a model of a 3D printer) up to 65.4% improvement of tracking accuracy using the proposed hybrid FBF compared to a standard FBF controller that does not incorporate data.

The outline of the paper is as follows: In Section II, an overview of the linear hybrid model from the authors' prior work is presented. Section III reviews the FBF controller and its receding horizon implementation and introduces the overall framework of the proposed hybrid FBF approach. Improved tracking performance using the proposed hybrid FBF controller is demonstrated via simulations in Section IV, followed by the conclusions and the future work in Section V.

II. Overview of the Linear Hybrid Model

Consider a stable and causal SISO dynamic system $\mathbf{H}$ containing linear dynamics $\mathbf{H}_L$ and unmodeled nonlinear dynamics $\mathbf{H}_{NL}$, as shown in Fig. 1. The system is sampled at interval T_s; $u(k)$ and $y(k)$, where $k = 0, 1, 2, \ldots$, are the input and output of $\mathbf{H}$ at time step k, respectively. Also, the inputs and outputs are fed into and measured from the system in small batches, denoted by the superscript (j), i.e.,

$$\boldsymbol{y}^{(j)} = \left[y(jN_p),\, y(jN_p+1),\, \ldots,\, y\left((j+1)N_p - 1\right) \right]^{\mathrm{T}}, \tag{1}$$

where N_p is the number of time steps in one batch and $j = 0, 1, 2, \ldots$ is the batch index.

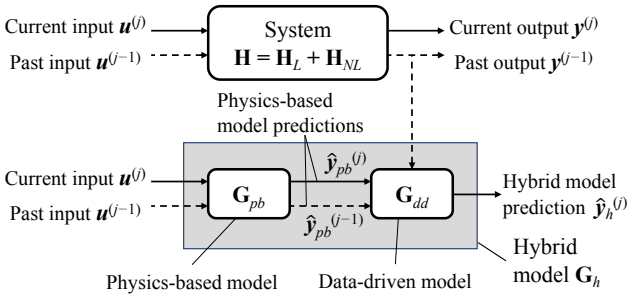

Fig. 1 General framework of the linear hybrid model [19]

Assume that $\mathbf{H}_L$ is accurately modeled by a physics-based model $\mathbf{G}_{pb}$, which can be a transfer function, a state space model, or a lifted system representation, with its prediction denoted by $\hat{\boldsymbol{y}}_{pb}$, while $\mathbf{H}_{NL}$ is unmodeled. Note that the hat accent on $\boldsymbol{y}$ means the prediction of the response by models. To account for the prediction error due to $\mathbf{H}_{NL}$, a more accurate hybrid model $\mathbf{G}_h$ was introduced in [19]. The hybrid model $\mathbf{G}_h$ linearly cascades the physics-based model $\mathbf{G}_{pb}$ and a data-driven model $\mathbf{G}_{dd}$ to enhance the prediction, as shown in Fig. 1. The data-driven model makes use of the past batch of system outputs (obtained online), as well as the past and the current predictions of $\mathbf{G}_{pb}$. Accordingly, the prediction of the linear hybrid model in the j-th batch, denoted as $\hat{\boldsymbol{y}}_h^{(j)}$, is given by

$$\begin{aligned} \hat{\boldsymbol{y}}_h^{(j)} &= \mathbf{G}_{dd}\left(\hat{\boldsymbol{y}}_{pb}^{(j-1)}, \hat{\boldsymbol{y}}_{pb}^{(j)}, \boldsymbol{y}^{(j-1)}\right) \\ &= \mathbf{G}_{dd}\left(\mathbf{G}_{pb}\boldsymbol{u}^{(j-1)}, \mathbf{G}_{pb}\boldsymbol{u}^{(j)}, \boldsymbol{y}^{(j-1)}\right) \\ &= \mathbf{G}_h\left(\boldsymbol{u}^{(j-1)}, \boldsymbol{u}^{(j)}, \boldsymbol{y}^{(j-1)}\right). \end{aligned} \tag{2}$$

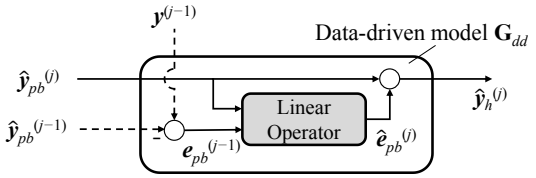

Fig. 2 Detailed structure of the data-driven model $\mathbf{G}_{dd}$ [19]

The detailed structure of $\mathbf{G}_{dd}$ is shown in Fig. 2. Its adoption of linear regression keeps $\mathbf{G}_h$ linear. The linear regression model aims to estimate the physics-based model prediction error $\boldsymbol{e}_{pb} \triangleq \boldsymbol{y} - \hat{\boldsymbol{y}}_{pb}$. Accordingly, in batch j, the prediction of $\mathbf{G}_h$ is recursively obtained by applying

$$\hat{y}_h(k) = \hat{y}_{pb}(k) + \hat{e}_{pb}(k) = \hat{y}_{pb}(k) + \hat{\boldsymbol{w}}^{(j)\mathrm{T}}\boldsymbol{\varphi}(k), \tag{3}$$

where the time step k is defined within the j-th batch as in Eq. (1), $\hat{\boldsymbol{w}}^{(j)}$ is the weight vector for the linear operator at the corresponding j-th batch, and $\boldsymbol{\varphi}(k)$ is the feature vector for time step k, which contains the elements in $\hat{\boldsymbol{y}}_{pb}^{(j-1)}$, $\hat{\boldsymbol{y}}_{pb}^{(j)}$ and $\boldsymbol{e}_{pb}^{(j-1)}$ (note that $\boldsymbol{e}_{pb}^{(j-1)}$ includes $\hat{\boldsymbol{y}}_{pb}^{(j-1)}$ and $\boldsymbol{y}^{(j-1)}$ due to $\boldsymbol{e}_{pb} \triangleq \boldsymbol{y} - \hat{\boldsymbol{y}}_{pb}$). Vector $\boldsymbol{\varphi}(k)$ is formulated as

$$\begin{aligned} \boldsymbol{\varphi}(k) = [1, \hat{y}_{pb}(k-q+1), \ldots, \hat{y}_{pb}(k), \\ e_{pb}(k-p), \ldots, e_{pb}(k-1)]^{\mathrm{T}}, \end{aligned} \tag{4}$$

where q and p are design parameters that determine the number of time steps of $\hat{\boldsymbol{y}}_{pb}$ and $\boldsymbol{e}_{pb}$ included in $\boldsymbol{\varphi}$. The selection of q and p depends on the dynamics of the system and the length of the prediction horizon. Note that e_{pb} requires measured data obtained online in batches; therefore, $e_{pb}(k)$ is not available for $k \geq jN_p$. Accordingly, $e_{pb}(k)$ in Eq. (4) is replaced by an estimated physics-based predicted error $\hat{e}_{pb}(k)$, as in Eq. (3), when $k \geq jN_p$. The training of $\hat{\boldsymbol{w}}^{(j)}$ is performed recursively by solving a regularized least squares optimization problem. It is designed to minimize the difference between the measured $e_{pb}(k)$ and the predicted $\hat{e}_{pb}(k)$ for all time steps before the current prediction batch using the following loss function

$$\hat{\boldsymbol{w}}^{(j)} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \sum_{k=0}^{jN_p-1} \left(e_{pb}(k) - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\varphi}(k)\right)^2 + \lambda\|\boldsymbol{w}\|_2^2, \tag{5}$$

where $\lambda > 0$ is the regularization factor that prevents overfitting. Note that Eq. (5) is solved using the analytical solution of ridge regression for $k = 0$ and is updated using recursive least squares (RLS) for $k > 0$.

III. Hybrid Filtered Basis Functions (FBF) Approach

This section briefly discusses the standard FBF approach [8], [9], followed by a detailed discussion of the proposed hybrid FBF approach.

A. Overview of the FBF Approach

Consider a linear system $\mathbf{H}_L$ aiming to track the desired reference trajectory $\boldsymbol{y}_d$. In the filtered basis function (FBF) controller, the input $\boldsymbol{u}$ is expressed as a linear combination of a set of basis functions $\boldsymbol{\psi}_i$ ($i = 0, 1, \ldots, n$), i.e.,

$$\boldsymbol{u} = \sum_{i=0}^{n} \gamma_i \boldsymbol{\psi}_i = \boldsymbol{\Psi}\boldsymbol{\gamma}, \tag{6}$$

where γ_i is the coefficient of the linear combination, $\boldsymbol{\Psi} = [\boldsymbol{\psi}_0, \boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_n]$ and $\boldsymbol{\gamma} = [\gamma_0, \gamma_1, \ldots, \gamma_n]^{\mathrm{T}}$. Due to the property of linearity, the output $\boldsymbol{y}$ is also a linear combination of the basis functions filtered through $\mathbf{G}_{pb}$, i.e.,

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on December 29,2021 at 16:07:02 UTC from IEEE Xplore. Restrictions apply.

$$\hat{\boldsymbol{y}} = \sum_{i=0}^{n} \gamma_i \tilde{\boldsymbol{\psi}}_i = \tilde{\boldsymbol{\Psi}}\boldsymbol{\gamma}, \tag{7}$$

where $\tilde{\boldsymbol{\psi}}_i = \mathbf{G}_{pb}\boldsymbol{\psi}_i$ are the filtered basis functions and $\tilde{\boldsymbol{\Psi}} = [\tilde{\boldsymbol{\psi}}_0, \tilde{\boldsymbol{\psi}}_1, \ldots, \tilde{\boldsymbol{\psi}}_n]$ is the filtered basis function matrix. The FBF method minimizes the tracking error $\boldsymbol{e}$, which is defined as

$$\min_{\boldsymbol{\gamma}} \|\boldsymbol{e}\|_2^2 \triangleq \min_{\boldsymbol{\gamma}} \|\boldsymbol{y}_d - \hat{\boldsymbol{y}}\|_2^2 = \min_{\boldsymbol{\gamma}} \|\boldsymbol{y}_d - \tilde{\boldsymbol{\Psi}}\boldsymbol{\gamma}\|_2^2, \tag{8}$$

where the optimal $\boldsymbol{\gamma}$ is obtained by the pseudoinverse of $\tilde{\boldsymbol{\Psi}}$

$$\boldsymbol{\gamma} = \tilde{\boldsymbol{\Psi}}^{\dagger}\boldsymbol{y}_d = \left(\tilde{\boldsymbol{\Psi}}^{\mathrm{T}}\tilde{\boldsymbol{\Psi}}\right)^{-1}\tilde{\boldsymbol{\Psi}}^{\mathrm{T}}\boldsymbol{y}_d, \tag{9}$$

and thereby the input to the system is $\boldsymbol{u} = \boldsymbol{\Psi}\boldsymbol{\gamma}$.

A limited-preview (or receding horizon) version of the FBF method using B-spline, called limited-preview filtered B-spline (LPFBS), was proposed in [9]. It partitions the trajectories into overlapping windows and then minimizes the tracking error window-by-window. The overlapping window, denoted by the superscript $[j]$, has length $N_w > N_p$, starting at the same time step as the non-overlapping batches described in Eq. (1), i.e.,

$$\boldsymbol{y}^{[j]} = \left[y(jN_p), y(jN_p+1), \ldots, y(jN_p+N_w-1)\right]^{\mathrm{T}}. \tag{10}$$

Note that the round bracket (j) in the superscript in Eq. (1) represents the non-overlapping batches for prediction, while the square bracket $[j]$ in Eq. (10) represents the overlapping window for control. Fig. 3 shows the differences between the batches and the windows.

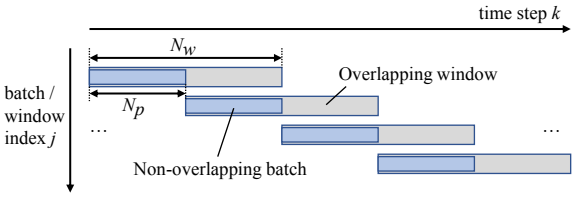

Fig. 3 The pictorial view of the definition of batches and windows

To implement LPFBS, $\hat{\boldsymbol{y}}$ is decomposed into past, current, and future sections with respect to the interested window j, i.e.,

$$\begin{aligned} \hat{\boldsymbol{y}}_P^{[j]} &= \left[y(0), y(1), \ldots, y(jN_p-1)\right]^{\mathrm{T}}, \\ \hat{\boldsymbol{y}}_C^{[j]} &= \left[y(jN_p), y(jN_p+1), \ldots, y(jN_p+N_w-1)\right]^{\mathrm{T}} = \hat{\boldsymbol{y}}^{[j]}, \\ \hat{\boldsymbol{y}}_F^{[j]} &= \left[y(jN_p+N_w), y(jN_p+N_w+1), \ldots\right]^{\mathrm{T}}, \end{aligned} \tag{11}$$

Accordingly, Eq. (7) is rewritten as

$$\begin{bmatrix} \hat{\boldsymbol{y}}_P^{[j]} \\ \hat{\boldsymbol{y}}_C^{[j]} \\ \hat{\boldsymbol{y}}_F^{[j]} \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{\Psi}}_P^{[j]} & \mathbf{0} & \mathbf{0} \\ \tilde{\boldsymbol{\Psi}}_{PC}^{[j]} & \tilde{\boldsymbol{\Psi}}_C^{[j]} & \mathbf{0} \\ \tilde{\boldsymbol{\Psi}}_{PF}^{[j]} & \tilde{\boldsymbol{\Psi}}_{CF}^{[j]} & \tilde{\boldsymbol{\Psi}}_F^{[j]} \end{bmatrix} \begin{bmatrix} \boldsymbol{\gamma}_P^{[j]} \\ \boldsymbol{\gamma}_C^{[j]} \\ \boldsymbol{\gamma}_F^{[j]} \end{bmatrix}. \tag{12}$$

where the subscripts P, C, F in $\tilde{\boldsymbol{\Psi}}$ respectively represent the past, current, and future input-output effects, while PC in $\tilde{\boldsymbol{\Psi}}$ means the effects of the past inputs on the current outputs; PF and CF are similarly defined. Also, since it is assumed that the system is causal, $\tilde{\boldsymbol{\Psi}}$ in Eq. (12) is a lower triangular matrix.

Unlike the full-preview version proposed in [8], the LPFBS minimizes the tracking error in every local window j, i.e., $\boldsymbol{e}^{[j]} = \boldsymbol{y}_d^{[j]} - \hat{\boldsymbol{y}}^{[j]}$ is to be minimized, where $\boldsymbol{y}_d^{[j]}$ represents the desired trajectory in the j-th window and $\hat{\boldsymbol{y}}^{[j]}$ is as defined in Eqs. (11) and (12). Thus, the objective is formulated as

$$\begin{aligned} \min_{\boldsymbol{\gamma}_C^{[j]}} \left\|\boldsymbol{e}^{[j]}\right\|_2^2 &\triangleq \min_{\boldsymbol{\gamma}_C^{[j]}} \left\|\boldsymbol{y}_d^{[j]} - \hat{\boldsymbol{y}}^{[j]}\right\|_2^2 \\ &= \min_{\boldsymbol{\gamma}_C^{[j]}} \left\|\boldsymbol{y}_d^{[j]} - \left(\tilde{\boldsymbol{\Psi}}_{PC}^{[j]}\boldsymbol{\gamma}_P^{[j]} + \tilde{\boldsymbol{\Psi}}_C^{[j]}\boldsymbol{\gamma}_C^{[j]}\right)\right\|_2^2, \end{aligned} \tag{13}$$

where $\boldsymbol{\gamma}_C^{[j]}$ are the coefficients for the current (j-th) window, $\boldsymbol{\gamma}_P^{[j]}$ represents all coefficients calculated before $\boldsymbol{\gamma}_C^{[j]}$, while $\tilde{\boldsymbol{\Psi}}_{PC}^{[j]}$ and $\tilde{\boldsymbol{\Psi}}_C^{[j]}$ are the corresponding operators, as defined in Eq. (12). Since $\boldsymbol{\gamma}_C^{[j]}$ is the only term in Eq. (13) that is unknown, it is computed by

$$\boldsymbol{\gamma}_C^{[j]} = \tilde{\boldsymbol{\Psi}}_C^{[j]\dagger}\left(\boldsymbol{y}_d^{[j]} - \tilde{\boldsymbol{\Psi}}_{PC}^{[j]}\boldsymbol{\gamma}_P^{[j]}\right), \tag{14}$$

And then the optimal control inputs for j-th window, $\boldsymbol{u}^{[j]}$, are given by

$$\boldsymbol{u}^{[j]} = \boldsymbol{\Psi}_{PC}^{[j]}\boldsymbol{\gamma}_P^{[j]} + \boldsymbol{\Psi}_C^{[j]}\boldsymbol{\gamma}_C^{[j]}. \tag{15}$$

For continuity, only the first N_p inputs from the total N_w are updated and fed into the system. Therefore, the input vector for the current batch, $\boldsymbol{u}^{(j)}$, is given by

$$\boldsymbol{u}^{(j)} = \begin{bmatrix} \mathbf{I}_{N_p} & \mathbf{0} \end{bmatrix} \boldsymbol{u}^{[j]}, \tag{16}$$

where $\mathbf{I}_{Np}$ represents the $N_p \times N_p$ identity matrix, and $\mathbf{0}$ is a zero matrix of appropriate dimension.

B. Implementation of Hybrid FBF Approach using Linear Hybrid Model

As mentioned in Sections I and II, the tracking performance of the FBF controllers may deteriorate when the unmodeled $\mathbf{H}_{NL}$ in the system is significant. Therefore, an FBF with the correction of measurement data is proposed. More specifically, the proposed FBF approach aims to enhance the LPFBS framework using the linear hybrid model. Since the proposed approach combines both physics-based and data-driven models, it is called hybrid FBF.

The general idea of the proposed hybrid FBF is to obtain a more accurate but still linear input-output relationship using the linear hybrid model $\mathbf{G}_h$. In other words, in addition to the linear model, the linear regression model is also used within the FBF framework. As in the LPFBS, the desired trajectory is partitioned into several overlapping windows of length N_w for optimization, but only the first N_p time steps are packaged in one batch and sent to the system for control. In contrast to the prediction aspect of the linear hybrid model that is performed recursively to obtain the predicted outputs, for control, the linear hybrid model requires a window-to-window mapping. That is, Eq. (2) and Eq. (3) are combined as

$$\hat{\boldsymbol{y}}_h^{[j]} = \hat{\boldsymbol{y}}_{pb}^{[j]} + \hat{\boldsymbol{e}}_{pb}^{[j]} = \mathbf{G}_{dd}\left(\hat{\boldsymbol{y}}_{pb}^{[j-1]}, \hat{\boldsymbol{y}}_{pb}^{[j]}, \boldsymbol{y}^{[j-1]}\right) = \mathbf{L}_{dd}^{[j]} \begin{bmatrix} 1 \\ \hat{\boldsymbol{y}}_{pb}^{[j-1]} \\ \hat{\boldsymbol{y}}_{pb}^{[j]} \\ \boldsymbol{e}_{pb}^{[j-1]} \end{bmatrix}, \tag{17}$$

where $\mathbf{L}_{dd}^{[j]}$ is the matrix representation of $\mathbf{G}_{dd}$ for the j-th window. The rows of $\mathbf{L}_{dd}^{[j]}$ are constructed based on $\hat{\boldsymbol{w}}^{(j)\mathrm{T}}$ in Eq. (3), and the details of constructing $\mathbf{L}_{dd}^{[j]}$ are presented in the Appendix.

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on December 29,2021 at 16:07:02 UTC from IEEE Xplore. Restrictions apply.

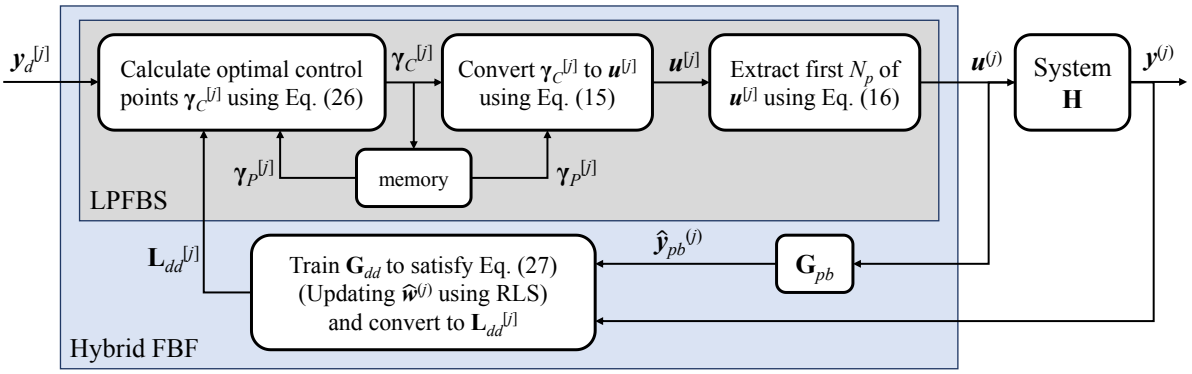

Fig. 4 Architecture of the proposed Hybrid FBF

Besides, $\hat{\boldsymbol{y}}_{pb}^{[j-1]}$ and $\boldsymbol{e}_{pb}^{[j-1]}$ are defined as

$$\hat{\boldsymbol{y}}_{pb}^{[j-1]} = \left[\hat{y}_{pb}(jN_p - q+1),\ \hat{y}_{pb}(jN_p - q+2),\ \ldots,\ \hat{y}_{pb}(jN_p - 1)\right]^{\mathrm{T}} \quad (18)$$

and

$$\boldsymbol{e}_{pb}^{[j-1]} = \left[e_{pb}(jN_p - p), e_{pb}(jN_p - p+1), \ldots,\ e_{pb}(jN_p - 1)\right]^{\mathrm{T}} \quad (19)$$

respectively; $\hat{\boldsymbol{y}}_{pb}^{[j]}$ is defined similar to $\hat{\boldsymbol{y}}_C^{[j]}$ in Eq. (11) as

$$\hat{\boldsymbol{y}}_{pb}^{[j]} = \left[\hat{y}_{pb}(jN_p),\ \hat{y}_{pb}(jN_p+1),\ \ldots,\ \hat{y}_{pb}(jN_p + N_w - 1)\right]^{\mathrm{T}}. \quad (20)$$

Furthermore, Eq. (17) is decomposed into an alterable part ($\hat{\boldsymbol{y}}_{pb}^{[j]}$) and an unalterable part ($\boldsymbol{\varphi}_u^{[j]}$) as

$$\hat{\boldsymbol{y}}_h^{[j]} = \mathbf{L}_{dd,a}^{[j]}\,\hat{\boldsymbol{y}}_{pb}^{[j]} + \mathbf{L}_{dd,u}^{[j]}\boldsymbol{\varphi}_u^{[j]}, \quad (21)$$

where $\mathbf{L}_{dd,a}^{[j]}$ and $\mathbf{L}_{dd,u}^{[j]}$ respectively represent the linear operators for alterable terms and the unalterable terms, extracted from the columns of $\mathbf{L}_{dd}^{[j]}$. Vector $\boldsymbol{\varphi}_u^{[j]}$ comprises all the unalterable terms for the j-th batch, i.e.,

$$\boldsymbol{\varphi}_u^{[j]} = \left[1 \quad \hat{\boldsymbol{y}}_{pb}^{[j-1]} \quad \boldsymbol{e}_{pb}^{[j-1]}\right]^{\mathrm{T}}. \quad (22)$$

Vector $\hat{\boldsymbol{y}}_{pb}^{[j]}$ in Eq. (21) is rewritten, according to Eq. (12), as

$$\hat{\boldsymbol{y}}_{pb}^{[j]} = \tilde{\boldsymbol{\Psi}}_{PC}^{[j]}\,\boldsymbol{\gamma}_P^{[j]} + \tilde{\boldsymbol{\Psi}}_C^{[j]}\,\boldsymbol{\gamma}_C^{[j]}, \quad (23)$$

where, as in Eq. (12), $\boldsymbol{\gamma}_C^{[j]}$ represents the coefficients that affect $\hat{\boldsymbol{y}}_{pb}$ starting from j-th window, $\boldsymbol{\gamma}_P^{[j]}$ represents the coefficients prior to $\boldsymbol{\gamma}_C^{[j]}$, while $\tilde{\boldsymbol{\Psi}}_{PC}^{[j]}$ and $\tilde{\boldsymbol{\Psi}}_C^{[j]}$ represent the linear effects of $\boldsymbol{\gamma}_P^{[j]}$ and $\boldsymbol{\gamma}_C^{[j]}$ on $\hat{\boldsymbol{y}}_{pb}^{[j]}$, respectively. Combining Eq. (21) and Eq. (23), the final prediction $\hat{\boldsymbol{y}}_h$ is given by

$$\hat{\boldsymbol{y}}_h^{[j]} = \mathbf{L}_{dd,a}^{[j]}\tilde{\boldsymbol{\Psi}}_{PC}^{[j]}\,\boldsymbol{\gamma}_P^{[j]} + \mathbf{L}_{dd,a}^{[j]}\tilde{\boldsymbol{\Psi}}_C^{[j]}\,\boldsymbol{\gamma}_C^{[j]} + \mathbf{L}_{dd,u}^{[j]}\,\boldsymbol{\varphi}_u^{[j]}. \quad (24)$$

To minimize the tracking error for the j-th window, $\boldsymbol{\gamma}_C^{[j]}$ is the only optimization variable and is used to minimize the objective function given by

$$\min_{\boldsymbol{\gamma}_C^{[j]}} \left\|\boldsymbol{e}^{[j]}\right\|_2^2 = \min_{\boldsymbol{\gamma}_C^{[j]}} \left\|\boldsymbol{y}_d^{[j]} - \hat{\boldsymbol{y}}_h^{[j]}\right\|_2^2. \quad (25)$$

The optimal control points $\boldsymbol{\gamma}_C^{[j]}$ are calculated as

$$\boldsymbol{\gamma}_C^{[j]} = \left(\mathbf{L}_{dd,a}^{[j]}\tilde{\boldsymbol{\Psi}}_C^{[j]}\right)^{\dagger}\left(\boldsymbol{y}_d^{[j]} - \mathbf{L}_{dd,a}^{[j]}\tilde{\boldsymbol{\Psi}}_{PC}^{[j]}\,\boldsymbol{\gamma}_P^{[j]} - \mathbf{L}_{dd,u}^{[j]}\,\boldsymbol{\varphi}_u^{[j]}\right), \quad (26)$$

and then the inputs for the j-th batch, $\boldsymbol{u}^{(j)}$, are calculated by the exactly same method as the LPFBS, i.e. by Eqs. (15) and (16).

With $\boldsymbol{u}^{(j)}$ fed into the system and the corresponding output $\boldsymbol{y}^{(j)}$ measured, the linear hybrid model is also updated by training $\hat{\boldsymbol{w}}$ recursively, i.e.,

$$\hat{\boldsymbol{w}}^{(j)} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \sum_{k=0}^{jN_p-1}\left(e_{pb}(k) - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\varphi}(k)\right)^2 + \lambda\|\boldsymbol{w}\|_2^2, \quad (27)$$

where $\boldsymbol{e}_{pb}$ is obtained by

$$\boldsymbol{e}_{pb}^{(j)} = \boldsymbol{y}^{(j)} - \hat{\boldsymbol{y}}_{pb}^{(j)} = \boldsymbol{y}^{(j)} - \left(\tilde{\boldsymbol{\Psi}}_{PC}^{(j)}\,\boldsymbol{\gamma}_P^{(j)} + \tilde{\boldsymbol{\Psi}}_C^{(j)}\boldsymbol{\gamma}_C^{(j)}\right). \quad (28)$$

The hybrid FBF approach is summarized in Fig. 4.

IV. Validation by Simulation

Consider a commercial vibration-prone 3D printer as in [9]. Due to the stepper motor dynamics and the flexible structure of the 3D printer, the print head can suffer from significant vibration, resulting in vibration marks on the printed parts [9]. Suppose the x- and y-axis of the 3D printer are fully decoupled; the vibrational dynamics in either axis can be modeled by a spring-mass-damper system shown in Fig. 5. The system experiences a disturbance force H_{NL} assumed to be created by unmodeled nonlinear guideway friction and stiffness from the printer's cabling system.

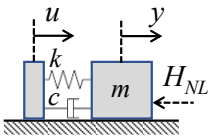

Fig. 5 Spring-mass-damper model representing the dynamics of a motion axis of a 3D printer with unmodeled nonlinearity from friction and cabling

The motion axis' dynamics can be written as

$$m\ddot{y} + c\dot{y} + ky + H_{NL}(y, \dot{y}) = c\dot{u} + ku, \quad (29)$$

Let us assume that $m = 1$ kg, $c = 15.7$ kg/s, and $k = 24674$ N/m such that, without considering nonlinearity, the natural frequency is 25 Hz and the damping ratio is 5%. Further, assume that the nonlinear disturbance force is

$$H_{NL}(y,\dot{y}) = 0.1\ \mathrm{sgn}(\dot{y})\ \dot{y}^2 + \mathrm{sgn}(y)\ y^2. \quad (30)$$

Accordingly, the combined nonlinear system dynamics is given by

$$m\ddot{y} + (c + 0.1|\dot{y}|)\dot{y} + (k + |y|)y = c\dot{u} + ku, \quad (31)$$

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on December 29,2021 at 16:07:02 UTC from IEEE Xplore. Restrictions apply.

The physics-based model $\mathbf{G}_{pb}$ does not consider the unmodeled nonlinearity H_{NL}. It is obtained from the transfer function $G_{pb}(s)$:

$$G_{pb}(s)=\frac{cs+k}{ms^2+cs+k} \quad . \tag{32}$$

$G_{pb}(s)$ is converted to lifted form (i.e., a discrete finite impulse response matrix), yielding $\mathbf{G}_{pb}$.

To compare the performance of the standard FBF and the hybrid FBF approaches, simulations are conducted by forcing the system in Eq. (31) to track the desired trajectories $\boldsymbol{y}_d$ shown in Fig. 6(a) and Fig. 7(a). With sampling time $T_s = 1$ ms, $\boldsymbol{y}_d$ in Fig. 6(a) is a discrete staircase trajectory which travels from 0 mm to 30 mm and then back to 0 mm with an increment of 10 mm, while $\boldsymbol{y}_d$ in Fig. 7(a) is a variable-frequency oscillatory trajectory which is extracted from one axis of a circular spiral trajectory formulated as

$$r = 5 + (\frac{5}{2\pi})\theta, \qquad 0 \le \theta \le 8\pi. \tag{33}$$

Both trajectories are subject to kinematic limits given by

$$\dot{y}_{d,lim} = 80 \text{ mm/s}, \quad \ddot{y}_{d,lim} = 8 \text{ m/s}^2, \quad \dddot{y}_{d,lim} = 1000 \text{ m/s}^3. \tag{34}$$

For both the standard FBF and the hybrid FBF, the batch length $N_p = 100$, the window length $N_w = 200$, and the basis functions are fifth-order B-spline basis functions with knot vector spacing equal to 10 (equivalent to 0.01s). For the linear hybrid model parameters, $q = 4$ such that it implicitly includes up to the jerk information, while $p = 50$, which is half of N_p.

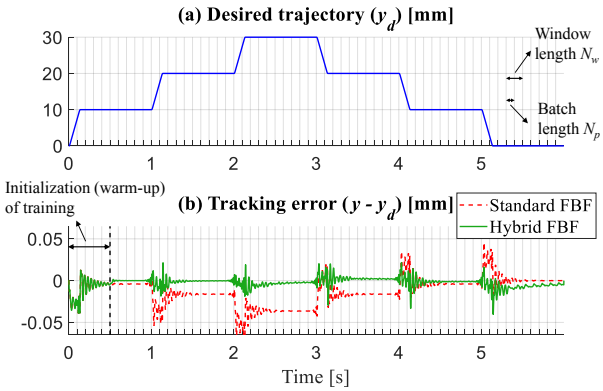

Fig. 6 (a) Desired trajectory (staircase); (b) Comparison of the tracking errors using the standard FBF and the hybrid FBF

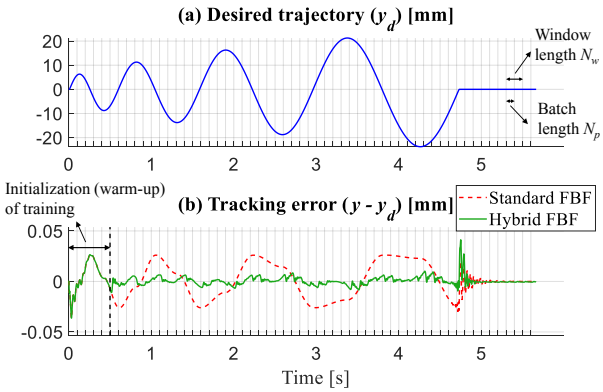

Fig. 7 (a) Desired trajectory (oscillatory); (b) Comparison of the tracking errors using the standard FBF and the hybrid FBF

Fig. 6(b) and Fig. 7(b) compare the tracking errors for both FBF approaches. It is shown that the tracking performance of the hybrid FBF approach is enhanced compared to the standard one when the system includes the nonlinearity $H_{NL}(y, \dot{y})$. In both simulations, the first five batches are employed for initializing the training of the hybrid FBF approach (i.e., warm-up). During the warm-up, $\mathbf{G}_{dd}$ is not used for control, so the hybrid FBF yields an identical response as the standard FBF. Beyond these initial training batches, the tracking errors of the hybrid FBF are significantly lower than the standard FBF in most portions. When abrupt changes or new characteristics occur in $\boldsymbol{y}_d$, the performance of the hybrid FBF deteriorates momentarily since $\mathbf{G}_{dd}$ has not been trained on the new operating conditions. However, it can quickly recover its performance upon training with new data. To sum up, the overall RMS of the tracking errors for the staircase and the oscillatory trajectories are respectively 20.3 μm and 16.8 μm for standard FBF, and are 7.0 μm and 6.1 μm for the hybrid FBF, i.e., 65.4 % and 63.8% reductions in RMS tracking error.

V. Conclusion and Future Work

This paper proposes a hybrid filtered basis functions (FBF) approach that enables the application of the FBF approach to feedforward tracking control of systems with unmodeled nonlinearity. Compared to the standard FBF, which only uses a physics-based model for control, the proposed hybrid FBF utilizes a linear hybrid model combining a physics-based model and a linear data-driven model. The data-driven model uses measured data to approximate unmodeled nonlinearity in a linear fashion.

In simulation case studies, the proposed hybrid FBF showed up to 65.4% improvements in tracking accuracy compared to the standard FBF because of its ability to approximate unmodeled nonlinearity from data. However, the performance of the hybrid FBF approach can degrade if the training of the data-driven model is insufficient. Thus, a warm-up period was required to allow for sufficient training. It is useful to determine the uncertainty of the hybrid model so as to optimally select the number of initialization batches. Also, to prevent transient loss of performance when operating conditions change abruptly, multiple data-driven models can be created to respectively model different nonlinear behaviors under various scenarios. Stability analysis and experimental validation of the hybrid FBF are also needed. These are subjects for future work.

Acknowledgment

This work is partially funded by the National Science Foundation through Grant # 1931950: CPS: Small: Mitigating Uncertainties in Computer Numerical Control (CNC) as a Cloud Service using Data-Driven Transfer Learning. A company founded by C.E. Okwudire holds a commercial license for the filtered B spline (FBS) algorithm.

Appendix

This appendix describes the conversion of the linear hybrid model from the recursive prediction, as expressed in Eq. (3) and (4), to the window-to-window mapping, as in Eq. (17).

Suppose the feature vector in Eq. (4) is partitioned into three different sections: the bias term, $\hat{\boldsymbol{y}}_{pb}$, and $\boldsymbol{e}_{pb}$. That is,

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on December 29,2021 at 16:07:02 UTC from IEEE Xplore. Restrictions apply.

$$\boldsymbol{\varphi}(k) = \begin{bmatrix} 1 & \hat{\boldsymbol{y}}_{pb}(k) & \boldsymbol{e}_{pb}(k) \end{bmatrix}^{\mathrm{T}}, \tag{35}$$

where $\hat{\boldsymbol{y}}_{pb}(k)$ includes $\hat{y}_{pb}(k-q+1)$, …, $\hat{y}_{pb}(k)$ and $\boldsymbol{e}_{pb}(k)$ includes $e_{pb}(k-p)$, …, $e_{pb}(k-1)$. Then, the corresponding weight $\hat{\boldsymbol{w}}$ in the $\hat{\boldsymbol{e}}_{pb}$ part in Eq. (3) can also be decomposed into the corresponding vectors as

$$\hat{\boldsymbol{w}} = \begin{bmatrix} \hat{\boldsymbol{w}}_{bias}, & \hat{\boldsymbol{w}}_{ypb}, & \hat{\boldsymbol{w}}_{epb} \end{bmatrix}^{\mathrm{T}}. \tag{36}$$

In order to construct a window-to-window mapping where the output contains all time steps in one window, say $\hat{y}_h(k)$ to $\hat{y}_h(k+N_w-1)$, the combined feature vector is given by

$$\begin{aligned} &[1, \hat{y}_{pb}(k-q+1), \ldots, \hat{y}_{pb}(k-1), \\ &\hat{y}_{pb}(k), \ldots, \hat{y}_{pb}(k+N_w-1), \\ &e_{pb}(k-p), \ldots, e_{pb}(k-1), \\ &\hat{e}_{pb}(k), \ldots \hat{e}_{pb}(k+N_w-2)]^{\mathrm{T}}, \end{aligned} \tag{37}$$

and the linear operator $\mathbf{K}_{dd}$ to perform the conversion, which is an $(N_w)\times(2N_w+p+q-1)$ matrix, is constructed as follows. For each row i, i.e., for $i = 1$ to N_w,

$$\begin{aligned} [\mathbf{K}_{dd}]_{i1} &= \hat{\boldsymbol{w}}_{bias} \\ [\mathbf{K}_{dd}]_{i(i+1:i+q)} &= \hat{\boldsymbol{w}}_{ypb} \\ [\mathbf{K}_{dd}]_{i(i+N_w+q:i+N_w+q+p-1)} &= \hat{\boldsymbol{w}}_{epb}, \end{aligned} \tag{38}$$

where $[\mathrm{A}]_{ij}$ denotes the element in the i-th row and the j-th column of the matrix A, and $[\mathrm{A}]_{i(j:n)}$ denotes the vector composed of the j-th to n-th elements in the i-th row of A.

However, since all $\hat{e}_{pb}$ terms in Eq. (37) are dependent on the other elements, so the independent elements only contain

$$\begin{aligned} &[1, \hat{y}_{pb}(k-q+1), \ldots, \hat{y}_{pb}(k-1), \\ &\hat{y}_{pb}(k), \ldots, \hat{y}_{pb}(k+N_w-1), \\ &e_{pb}(k-p), \ldots, e_{pb}(k-1)]^{\mathrm{T}}, \end{aligned} \tag{39}$$

which corresponds to the first N_w+p+q elements in Eq. (37) Therefore, the linear operator for Eq. (39), denoted as $\mathbf{L}_{dd}$, is rearranged as follows. Firstly, assign for each row i of $\mathbf{L}_{dd}$,

$$[\mathbf{L}_{dd}]_{i:} = [\mathbf{K}_{dd}]_{i(1:N_w+q+p)}, \tag{40}$$

where $[\mathrm{A}]_{i:}$ denotes the i-th row of A. Then, for each additional element between Eqs. (37) and (39), i.e., for $j = 1$ to N_w-1, recursively perform

$$\mathbf{L}_{dd} = \mathbf{L}_{dd} + [\mathbf{K}_{dd}]_{:(j+N_w+q+p)} \cdot [\mathbf{L}_{dd}]_{j:}, \tag{41}$$

where $[\mathrm{A}]_{:j}$ denotes the j-th column of A. Lastly, since in Eq. (3), additional term $\hat{y}_{pb}$ is added, the corresponding elements of $\mathbf{L}_{dd}$ are further added by 1, i.e., for $k = 1$ to N_w

$$[\mathbf{L}_{dd}]_{k(k+q)} = [\mathbf{L}_{dd}]_{k(k+q)} + 1. \tag{42}$$

The $\mathbf{L}_{dd}$ obtained in Eq. (42), and the feature vector in Eq. (39) are corresponding to the terms in Eq. (17).

REFERENCES

[1] G. M. Clayton, S. Tien, K. K. Leang, Q. Zou, and S. Devasia, "A review of feedforward control approaches in nanopositioning for high-speed SPM," *Journal of Dynamic Systems, Measurement and Control*, vol. 131, no. 6, 2009, doi: 10.1115/1.4000158.

[2] B. P. Rigney, L. Y. Pao, and D. A. Lawrence, "Nonminimum phase dynamic inversion for settle time applications," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 989–1005, 2009, doi: 10.1109/TCST.2008.2002035.

[3] J. van Zundert and T. Oomen, "On inversion-based approaches for feedforward and ILC," *Mechatronics*, vol. 50, pp. 282–291, 2018, doi: 10.1016/j.mechatronics.2017.09.010.

[4] M. Tomizuka, "Zero phase error tracking algorithm for digital control," *Journal of Dynamic Systems, Measurement and Control*, vol. 109, no. 1, pp. 65–68, 1987, doi: 10.1115/1.3143822.

[5] J. A. Butterworth, L. Y. Pao, and D. Y. Abramovitch, "Analysis and comparison of three discrete-time feedforward model-inverse control techniques for nonminimum-phase systems," *Mechatronics*, vol. 22, no. 5, pp. 577–587, 2012, doi: 10.1016/j.mechatronics.2011.12.006.

[6] D. Torfs, J. De Schutter, and J. Swevers, "Extended bandwidth zero phase error tracking control of nonminimal phase systems," *Journal of Dynamic Systems, Measurement and Control*, vol. 114, no. 3, pp. 347–351, 1992, doi: 10.1115/1.2897354.

[7] J. T. Wen and B. Potsaid, "An experimental study of a high performance motion control system," in *Proceedings of the 2004 American Control Conference*, 2004, pp. 5158–5163. doi: 10.23919/ACC.2004.1384671.

[8] K. S. Ramani, M. Duan, C. E. Okwudire, and A. G. Ulsoy, "Tracking control of linear time-invariant nonminimum phase systems using filtered basis functions," *Journal of Dynamic Systems, Measurement and Control*, vol. 139, no. 1, 2017, doi: 10.1115/1.4034367.

[9] M. Duan, D. Yoon, and C. E. Okwudire, "A limited-preview filtered B-spline approach to tracking control – With application to vibration-induced error compensation of a 3D printer," *Mechatronics*, vol. 56, pp. 287–296, 2018, doi: 10.1016/j.mechatronics.2017.09.002.

[10] K. S. Ramani, M. Duan, C. E. Okwudire, and A. G. Ulsoy, "Optimal selection of basis functions for minimum-effort tracking control of nonminimum phase systems using filtered basis functions," *Journal of Dynamic Systems, Measurement, and Control*, vol. 141, no. 11, 2019, doi: 10.1115/1.4044355.

[11] K. S. Ramani and C. E. Okwudire, "Optimal selection of basis functions for robust tracking control of uncertain linear systems – With application to three-dimensional printing," *Journal of Dynamic Systems, Measurement, and Control*, vol. 143, no. 10, 2021, doi: 10.1115/1.4051097.

[12] K. S. Ramani and C. E. Okwudire, "Two-stage robust tracking controller for linear systems with known uncertainty using filtered basis functions," in *Proceedings of the ASME 2020 Dynamic Systems and Control Conference*, vol. 2, 2020, doi: 10.1115/DSCC2020-3207.

[13] A. Karpatne, G. Atluri, J. H. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Samatova, V. Kumar, "Theory-guided data science: A new paradigm for scientific discovery from data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 10, pp. 2318–2331, 2017, doi: 10.1109/TKDE.2017.2720168.

[14] J. Willard, X. Jia, S. Xu, M. Steinbach, and V. Kumar, "Integrating physics-based modeling with machine learning: A survey," *arXiv*. 2020, ArXiv: abs/2003.04919.

[15] T. C. Tsao and M. Tomizuka, "Adaptive zero phase error tracking algorithm for digital control," *Journal of Dynamic Systems, Measurement and Control*, vol. 109, no. 4, pp. 349–354, 1987, doi: 10.1115/1.3143866.

[16] F. Boeren, T. Oomen, and M. Steinbuch, "Iterative motion feedforward tuning: A data-driven approach based on instrumental variable identification," *Control Engineering Practice*, vol. 37, pp. 11–19, 2015, doi: 10.1016/j.conengprac.2014.12.015.

[17] A. Dumanli and B. Sencer, "Data-driven iterative trajectory shaping for precision control of flexible feed drives," *IEEE/ASME Transactions on Mechatronics*, 2020, doi: 10.1109/TMECH.2020.3045444.

[18] W. Huang, K. Yang, Y. Zhu, and S. Lu, "Data-driven parameter tuning for rational feedforward controller: Achieving optimal estimation via instrumental variable," *IET Control Theory & Applications*, vol. 15, no. 7, pp. 937– 948, 2021, doi: 10.1049/cth2.12093.

[19] C.-H. Chou, M. Duan, and C. E. Okwudire, "A linear hybrid model for enhanced servo error pre-compensation of feed drives with unmodeled nonlinear dynamics," *CIRP Annals - Manufacturing Technology*, 2021, doi:10.1016/j.cirp.2021.04.070.

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on December 29,2021 at 16:07:02 UTC from IEEE Xplore. Restrictions apply.